



# Probabilistic Grid-based Collision Risk Prediction for Driving Application

Lukas Rummelhard, Amaury Nègre, Mathias Perrollaz, Christian Laugier

## ► To cite this version:

Lukas Rummelhard, Amaury Nègre, Mathias Perrollaz, Christian Laugier. Probabilistic Grid-based Collision Risk Prediction for Driving Application. ISER, Jun 2014, Marrakech/Essaouira, Morocco. hal-01011808

**HAL Id: hal-01011808**

**<https://inria.hal.science/hal-01011808>**

Submitted on 9 Jul 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Probabilistic Grid-based Collision Risk Prediction for Driving Application

Lukas Rummelhard<sup>1</sup>, Amaury Nègre<sup>2</sup>, Mathias Perrollaz<sup>1</sup>, and Christian Laugier<sup>1</sup>

<sup>1</sup> e-Motion, Inria Grenoble Rhône Alpes  
surname.name@inria.fr

<sup>2</sup> Laboratoire d'Informatique de Grenoble, CNRS  
surname.name@imag.fr

**Abstract.** In the recent years, more and more modern cars have been equipped with perception capabilities. One of the key applications of such perception systems is the estimation of a risk of collision. This is necessary for both Advanced Driver Assistance Systems and Autonomous Navigation. Most approach for risk estimation propose to detect and track the dynamic objects in the scene. Then the risk is estimated as a Time To Collision (TTC) by projecting the object's trajectory in the future. In this paper, we propose a new grid-based approach for collision risk prediction, based on the Hybrid-Sampling Bayesian Occupancy Filter framework. The idea is to compute an estimation of the TTC for each cell of the grid, instead of reasoning on objects. This strategy avoids to solve the difficult problem of multi-objects detection and tracking and provides a probabilistic estimation of the risk associated to each TTC value. After promising initial results, we propose in this paper to evaluate the relevance of the method for real on-road applications, by using a real-time implementation of our method in an experimental vehicle.

**Keywords:** Dynamic Occupancy Grid, Risk assessment, Time To Collision, Intelligent Vehicle

## 1 Introduction

In the development of ADAS (Advanced Driver Assistance Systems) that can prevent collision to append, a basic requirement is an assessment of the collision risk. This paper presents an original method for collision risk estimation. The idea is to compute an estimation of the TTC for each cell of a grid, instead of reasoning on objects. This strategy avoids to solve the difficult problem of multi-objects detection and tracking and provides a probabilistic estimation of the risk associated to each TTC value.

The method has been implemented for real time operation on GPU in our experimental vehicle. Then the vehicle is used for real on-road testing in order to evaluate the method in real conditions.

The paper is organized as follow: Section II presents related work. Section III describes the method. Section IV provides experimental results. Section V concludes.

## 2 Related work

In the literature, various approaches have been proposed in order to detect potential collisions in advance and trigger an assistance like a driver alarm or an automatic braking. Most of them rely on the detection and tracking of the moving objects in the scene. This tracking is used to estimate a risk metric, used for a decision layer. The simplest approach consists in computing a simple Time to Collision (TTC) by predicting the trajectories of both the ego-vehicle and the other objects using a simple linear motion model [1]. More advanced approach propose to generate a set of possible trajectories for all the objects, including the ego-vehicle, and compute the number of possible intersections between those trajectories [2]. The trajectory generation is performed by applying a set of possible controls to the objects. This approach has the advantage of providing a probabilistic estimation of the risk.

Considering that a control can change over time and is dependant from the intention of the driver, the authors in [3] propose to constrain the possible trajectories using the estimated maneuver intention of the driver. This allows to consider longer time prediction.

All this methods rely on the detection and tracking of the moving objects [4]. This stage is a difficult problem to solve, which can be computationally costly and can generate errors in the process. Alternatively to methods based on objects trajectory prediction, other methods propose to estimate time to collision based on visual features [5], but such methods are only designed to work with computer vision.

Another alternative to the object-based approaches are the grid-based approaches, like the Bayesian Occupancy Filter [6], or its extension Hybrid - Sampling Bayesian Occupancy Filter (HSBOF)[7], which represents the environment as a probabilistic occupancy grid. This framework allows to model both the static and the dynamic environment, by estimating velocity probability distributions for each cell in the grid. It is efficient for short term prediction, but since there are no notions of objects or behaviors, longer term prediction requires to integrate additional prior knowledge like map information [8].

Grid based methods are designed to be very efficiently implemented on parallel architecture and avoid to deal with object-level data association for both tracking and sensor fusion. Consequently, they are able to provide in real time a robust representation of the environment. However, in order to use such approaches for driving applications, the authors generally add a subsequent clustering stage to retrieve the notion of objects before any application [9, 3].

An hybrid approach is presented in [10], with *velocity obstacles*. The authors combine a notion of objects with an estimation of velocities within a dynamic occupancy grid.

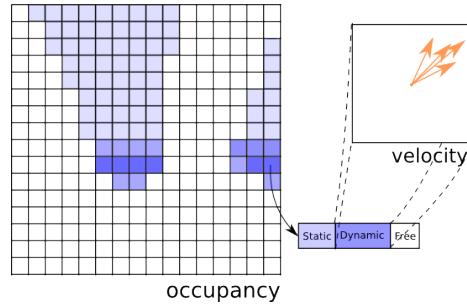
Based on the occupancy grid framework, we propose in this paper a method that allows to estimate both a probabilistic collision risk and a TTC for each cell of a grid, without any detection and tracking of objects. This dense representation of the risk is intended to be used to take decisions about the short

term evolution of the scene. In this work, we present the approach and propose to evaluate how it can be used for automotive applications.

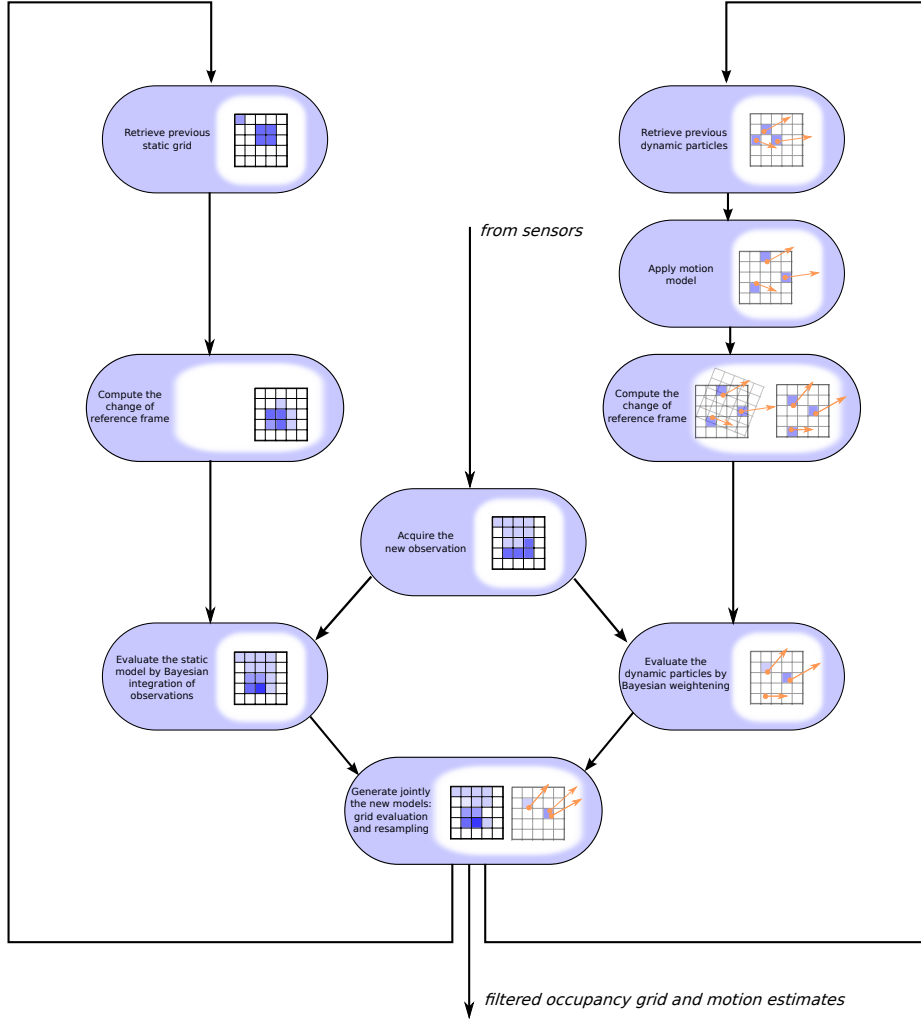
### 3 Description of the method

#### 3.1 HSBOF presentation

The Hybrid Sampling Bayesian Occupancy Filter (HSBOF)[7] is a Bayesian filtering perception technique which models the environment at a sub-object level, in term of spatial occupancy and dynamics. The surrounding of the subject is divided into cells, to which are associated random variables, symbolizing their occupancies and velocities. The scene is interpreted through the estimated distributions of those variables, estimations that are recursively updated according to the observations. The motion field of the scene is inferred through the spatial occupancy evolution, and is described as the combination of a static part, depicted by likelihood values in a regular grid, and a dynamic part, sampled by moving particles which transfer occupancy between cells. All the distributions are jointly generated and updated. The distinction in dynamics allows to apply specific motion representations, and then to efficiently allocate memory and computation power. Indeed, while the static part of the scene does not require an important number of samples to symbolize its velocity distribution, the dynamic one necessitate a substantial precision and reactivity. The figure 1 summarizes the HSBOF scene representation. This data structure also allows to conveniently change the reference frame, which is needed in case of a moving subject. Using motion sensors embedded in the mobile robot (Inertial Measurement Unit, GPS, Wheel speed and steering sensor, visual odometry, etc.), the displacement of the grid between two updates is estimated, then applied on the models. The global framework of the HSBOF is summarized on the figure 2.



**Fig. 1.** HSBOF data representation. A dense grid is used to represent the occupancy while for each cell velocity is split among a static field and a set of particles.



**Fig. 2.** HSBOF algorithm summary. From sensor data instantaneous occupancy grids are successively computed. Those observations are integrated in a Bayesian filter in which coexist and jointly adapt two models, a static grid and a dynamic set of moving particles. The result is obtained by their combination, which provides a filtered occupancy grid as well as inferred motion distributions for cells.

### 3.2 Grid-based collision assessment

As an output of the HSBOF, are available estimates of the robot surrounding occupancy and dynamics. Thanks to the already mentioned embedded motion sensors, the ego-motion is estimated. Using those and a robot model, which specifies its form, size and motion model, a time projection is achieved to assess collision risk in the future and localize it in the grid. Indeed, to each current

occupancy component is associated a velocity and motion model, in such a way that its position over time can be predicted. Simultaneously, the space occupied by the robot over time can also be assessed. The risk evaluation consists in a succession of scene configuration and robot position predictions, and so potential overlaps, in other words collisions can be assessed in time and space (figure 3).

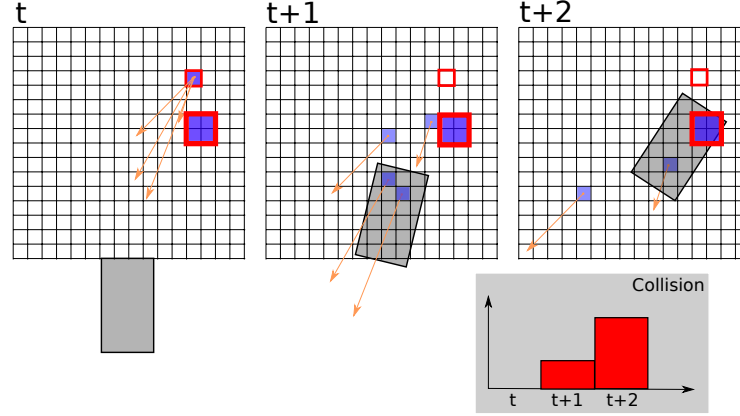
In practice, to each static cell and dynamic particle of the HSBOF representation are associated a "Time To Contact" (TTC) value and a risk weight. Until a time horizon, both vehicle and elementary components are iteratively moved according to their motion models. The time step used is chosen according to the grid resolution and the maximal reachable speed, to prevent any failed collision detection due to time resolution. The intersection between the vehicle and each basic component is evaluated. If a collision is confirmed, the TTC value is set and a weight is associated to the element, according to its associated quantity of occupancy. Otherwise, the next time step is processed. Beyond the chosen time range, if no collision is observed, a null weight is assigned (since there is no predicted collision, the TTC value is meaningless, by convention it is set to the time limit). Those computations performed, for each cell in the current occupancy grid a risk value over the time range is calculated, corresponding to the likelihood of the cell to be occupied, and for its content to collide with the robot during the selected period. The risk calculation simply consists in a sum over this time range of the related weights.

As a result at this stage, a risk grid presents for each cell the probability to hit an obstacle coming from that cell over a given period if the scene dynamics remain unchanged (conservative model, relevant for short-term prediction). Given that grid, a single global collision value is computed, and used to inform the system of any imminent danger. Two approaches are proposed to evaluate this general risk :

The first one proposes to evaluate the likelihood of colliding with a selected surface size. According to this area, and to the area of a cell, the number of  $n$  cells to be struck is thus given, the computed risk value being the probability to collide with  $n$  cells of the grid. This evaluation presents the advantage of rightfully integrate risk when the occupancy estimation is diffuse. However, in the case of high speed robot displacement, the integration of occluded spaces, yet with unknown occupancy, can lead to systematic high risk scores. Although crossing unknown areas is rightfully dangerous, the estimation becomes meaningless under such circumstances.

The second evaluation selects, after applying a median filter on the risk grid to remove noisy data, the maximum value of collision over the cells. This simpler approach turns out to be more effective in practice.

By comparing this value to a threshold, a risk detector is thus defined. The system is then used to generate alerts, for the driver if the timing makes it relevant, and for a driving assistance device.



**Fig. 3.** Collision risk estimation over time for a specific cell. The cell position is predicted according to its velocity, along with the mobile robot. This risk profile is computed for every cell, and then used to integrate over time the global collision risk.

## 4 Experimental Results

### 4.1 Experimental platform

For the experiments, we equipped a Lexus LS600h car with two Ibeo Lux multilayer lidars under the two front lights (see Fig.4). The horizontal field of view covered by the two lidars is almost 160 degrees. For the odometry calculation we use vehicle velocity and steering data collected from the CAN bus system. The input of our algorithm is a sequence of occupancy grids computed from the two lidars. To merge the eight laser scan layers acquired by the two lidars and to compute the input observation occupancy grid, we use a method similar to [11]. The vehicle is used to collect real on-road data for both downtown and highway scenarios, in order to assess the performances of the proposed approach.



**Fig. 4.** Experimental platform : Lexus LS600h car equipped with two Ibeo Lux Lidars and cameras.

The collected data are also processed in real time using a computer embedded in the car trunk. As the method is designed to be efficiently implemented on parallel, it is implemented on GPU, using nvidia CUDA. The grid resolution used is of 750x300, and the number of particles is 262144 (0.86 particle/ cell on average). The complete chain for perception (including lidar to grid mapping, HSBOF and TTC estimation) runs in real time at 15 FPS on a nVidia Quadro 2000 graphic board.

## 4.2 Methodology

The experiments focus on validating the approach in different contexts. On the one side, real on-road data is used to assess the realism of the approach in complex scenarios. On the other side, it is not possible to generate real risky situations on open road. Therefore, collision are performed with soft objects in a closed environment, to simulate risky driving situations.

Beyond the various contexts, we propose to use our risk estimation strategy for various applications. Therefore, we consider 3 different time horizons corresponding to 3 different TTC values:

- A1 - pre-crash: a collision is likely to appear in less than  $t_1 = 500ms$ . In this context, the driver is no longer able to react and the collision detection is used to prepare the vehicle to the impact;
- A2 - collision mitigation: a collision is likely to appear between  $t_1$  and  $t_2 = 1s$ . In this case, the vehicle can perform an automatic action (e.g. automatic braking) in order to mitigate the danger or avoid the impact;
- A3 - dangerous driving: a collision is likely to appear between  $t_2$  and  $t_3 = 1.5s$ . In this last case, the assistance system can warn the driver of a potential risk and the driver may react in a way or another

## 4.3 First experiment : dynamic collisions

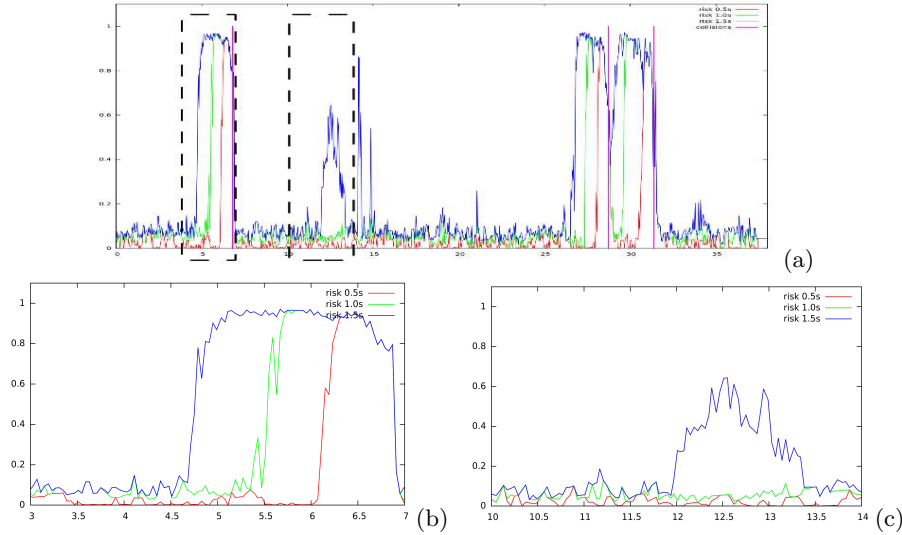
The first set of experiments was performed in a mostly controlled environment. On a parking lot, the experimental vehicle was used to model its environment and predicts potential collisions. To simulate obstacles, 3 bouncy balls of 55, 65 and 75 centimeter of diameter were thrown around the vehicle and on its path. The main advantage of the use of balls is the immediate validation of the existence of a risk. Indeed, on real road data, as actual collisions are not conceivable, the ground truth for risk is a hard task as it implies a subjective assessment. In this experiment, the vehicle could hit the obstacles, the collisions being annotated, confirming a preceding risk without ambiguity. Many scenarios were staged, including:

- immobile or moving vehicle
- immobile or moving obstacles
- various number of obstacles
- various trajectory options



- impacts or close crossings
- limited visibility (late appearance of obstacles, occlusions, intersection during curves)

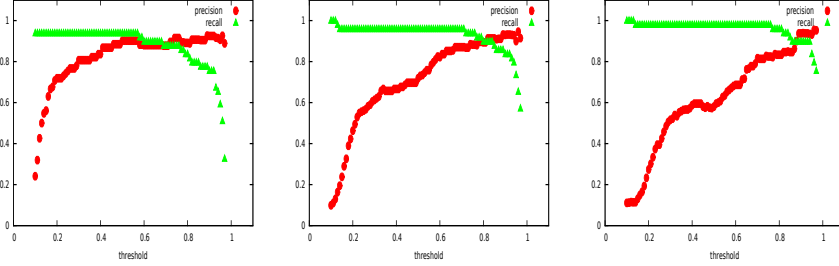
**Estimated risk values** Figure 5-a shows an example of the estimated risk values while the balls are thrown toward the vehicle. The impacts are annotated as the vertical purple lines. It appears that for all the impacts, a high risk value is properly predicted in advance. Figure 5-b shows in more detail an example of risk prediction before the impact. As expected, the risk corresponding to  $TTC \leq 1.5s$  rises first (between 1.5s and 2s before the collision), followed by the risk values corresponding to  $TTC \leq 1.0s$ , and  $TTC \leq 0.5s$ . Figure 5-c shows an example where the ball crosses close to the car without hitting it. In this case, the risk corresponding to  $TTC \leq 1.5s$  is high and then decreases as soon as the uncertainty about a possible collision becomes very low.



**Fig. 5.** (a) Evolution of the collision risk for  $TTC \leq 0.5s$ ,  $1.0s$ ,  $1.5s$ , in the case of multiple balls thrown toward the vehicle. (b) detail when a collision happens. (c) detail in case of the ball passing very near without collision. For  $1.5s$ , a significant risk is estimated, while for shorter  $TTC$  nothing is detected, as expected.

**Alerts generation** The experiment with the balloons has been repeated about a hundred times, including cases where the car hits the balloons and other where the balls come very close to the car without hitting it. This provides the opportunity to evaluate possible thresholds for generating alerts for the different values

of TTC. Figure 6 shows how the precision and recall rates change with respect to the threshold.



**Fig. 6.** Precision (red dots) and recall (green triangles) of risk detection according to the threshold selected. From left to right, those are given for a estimation over 0.5s, 1s and 1.5s.

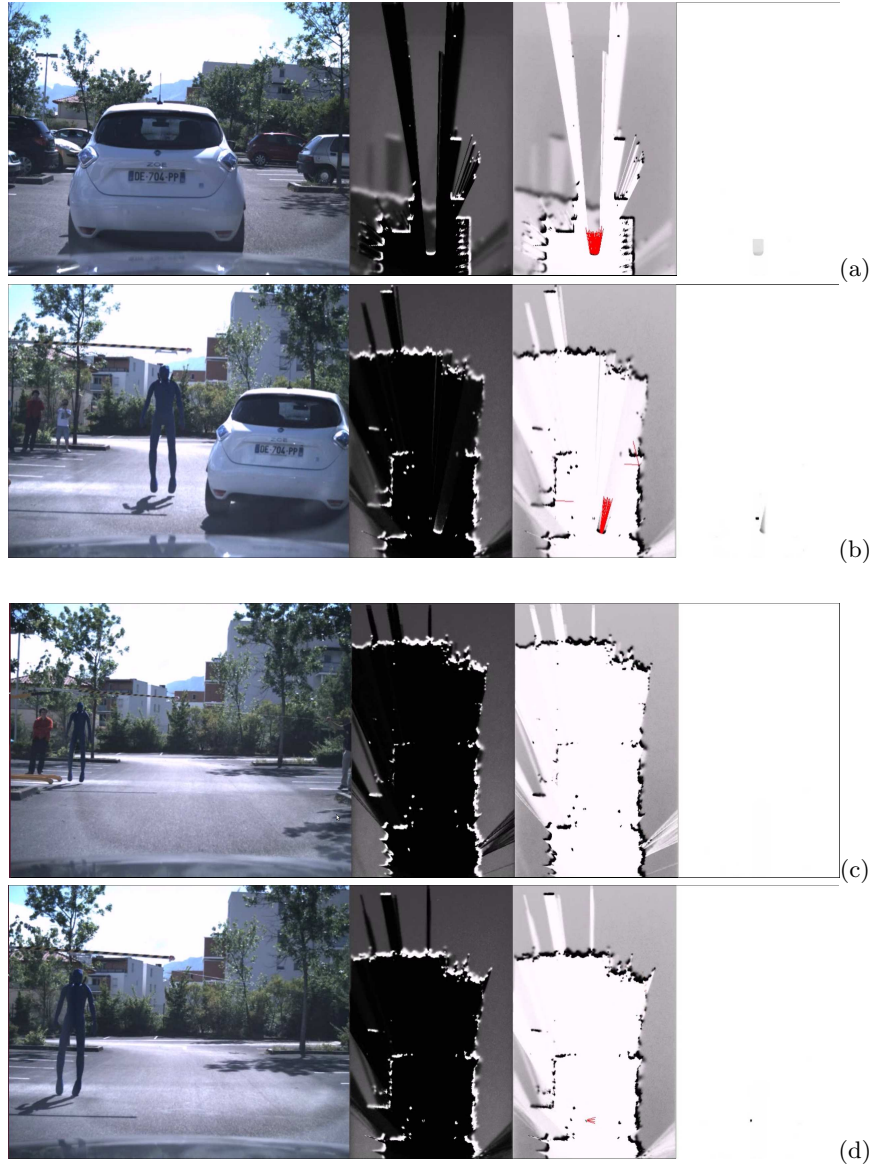
It appears on the precision/recall curves that efficient detectors can be made by selectionning the adequate threshold for every time horizon.

#### 4.4 Second experiment : the mannequin

In order to go further more realistic scenarios, we designed a fake pedestrian in the form of a tissue mannequin filled with bubble wrap. This allows to make collisions with a human-shaped objects. Figure 7 shows the mannequin and the system used for hanging it. This system is equipped with a runner, in order to permit lateral displacements of the pedestrian.

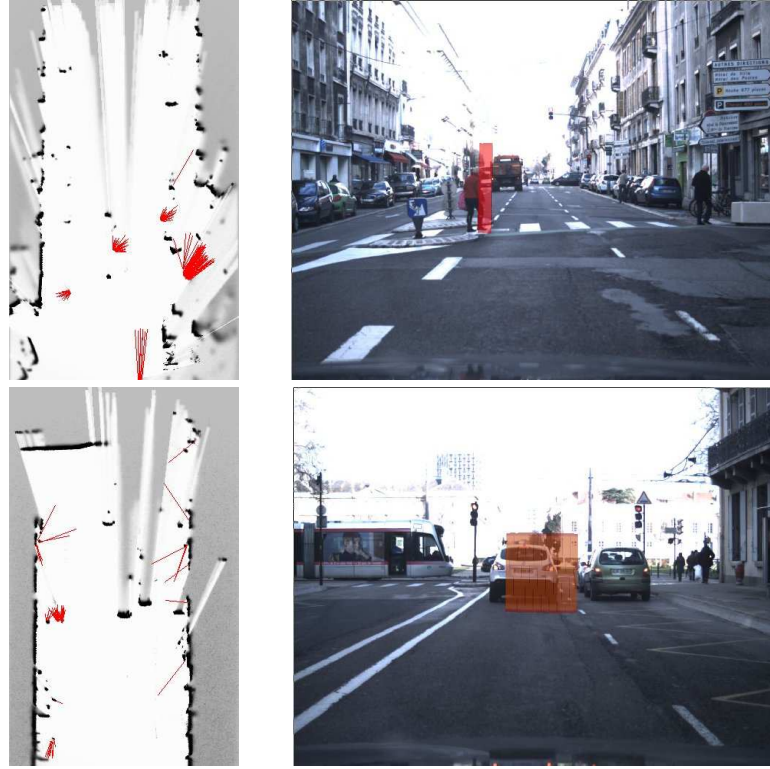


**Fig. 7.** (a) Fake pedestrian used for experiments. (b) The mannequin is attached to a system with a runner, in order to allow lateral displacements.



**Fig. 8.** Results of the system. Are presented each line a visual capture from the embedded camera, the estimated occupancy grid in front of the car (white for occupied, grey for unknown, black for empty), the estimated motion field (if a case is seen as dynamic, a red motion vector showing the average velocity in the cell is drawn on the map) and finally the estimated risk map for 0.5s. The first sequence (a) (b) presents the appearance of an occluded pedestrian, the second (c) (d) a moving pedestrian heading towards the road.

Over the different scenarios of car velocity, pedestrian motion and occlusion experimented, the system successfully generated risk alarms every time. Figure 8 shows examples of those experiments with the mannequin. The first examples corresponds to the sudden appearance of a pedestrian which was hidden by the preceding car. The risk map then shows immediately a important risk at the position of the pedestrian. It means that the system is reactive enough to detect a risk appearing at the last moment before the collision. The second one corresponds to the road crossing of a moving pedestrian. As the system estimate its motion, a risk appears on the risk map even before the pedestrian is on the car trajectory.

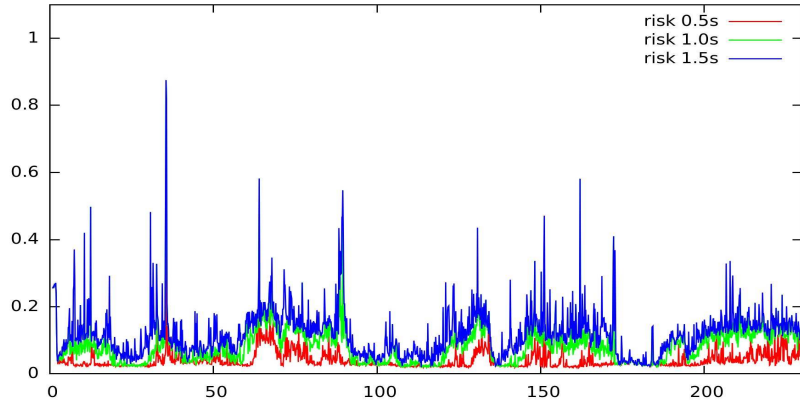


**Fig. 9.** First HSBOF and risk estimation results. Left image: output of the HSBOF process with the filtered occupancy grid (white for empty space, black for occupied space) and the velocity field (from each moving cell are drawn red rays representing the velocities). Right image: dangerous cells are reprojected in the camera image.

#### 4.5 Third experiment: On-road evaluation

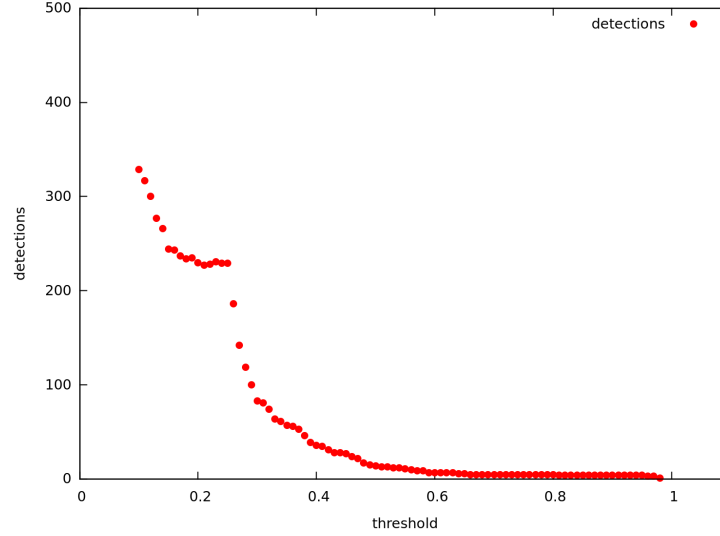
**HSBOF and risk estimation** We have performed experiments on various recorded sequences of road journey, in highway and downtown environments. Figure 9 shows the output of the HSBOF and the extraction of cells with a high short-term probability of collision. On the first example, the driver arrives too quickly at the traffic light, the car ahead then generates a collision risk corresponding to a A3 alert. On the second example, we can see that the system rightly detects the risk of collision with the pedestrian crossing the road, even though he is not already on the trajectory of the vehicle. Indeed, as the occupancy dynamics are inferred by the HSBOF, the method predicts the likely impact.

Figure 10 shows an example of the risk values estimated during a urban driving sequence. Since we are driving safely, the risk corresponding to  $TTC \leq 1.0s$  remains very low. There are a few peaks for the risk corresponding to  $TTC \leq 1.5s$ . These happen at the end of turning maneuvers where the predicted trajectory of the car is supposed to be circular, while the driver is about to go back to a straight trajectory. Thus, these cases do not generate high values of risk for shorter TTC values.



**Fig. 10.** Example of a sequence of driving downtown.

**Alarms generation on-road** In order to evaluate the relevance of the approach for pre-crash applications, we focus on the A1 alerts. The car is driven in various road environment (downtown, highway, national roads) to collect data. A dataset containing 27000 frames, including complex situations is recorded and the algorithm is applied, with a variable alarm threshold. The curve on figure 11 shows the number of A1 alarms with respect to the risk threshold. With a risk threshold set to 0.8 (which is relevant with respect to the curves shown on figure 6) only 4 alarms were generated.



**Fig. 11.** Alarms generated for risky situations at  $TTC \leq 0.5s$  while driving in crowded urban situations, with respect to the risk threshold (tested on 27000 frames).

## 5 Conclusion

In this paper, we presented a method for short-term collisions-risk estimation. The method takes advantage of the ability of the HSBOF algorithm to provide both an occupancy grid and accurate estimation of velocities over an occupancy grid. The use of this information, combined with the prediction of the vehicle's trajectory, allows to estimate Time To Collision for each cell of the grid. The method shows several advantages, like the ability to have probabilistic estimation of collision risk, the ability to be implemented efficiently in parallel, and also to estimate TTC in highly dynamic environment, without any object tracking algorithm.

A comprehensive evaluation of the algorithm has been proposed, based on both real road data and controlled situation. This evaluation showed that the method was able to perform well. In controlled environments the system is very reactive and every collision is properly detected while in difficult scenarios the system does not generate abusive alarms. There is still some work to do to deal with the remaining errors and to validate more extensively the algorithm on hundreds of kilometers, but the results obtained here from the experiments are promising. Therefore, we project to couple this perception layer with prototypes of Human-Machine-Interfaces for testing their use in Advanced Driver Assistance Systems.

## Acknowledgment

The authors would like to thank Toyota Motor Europe for their collaboration and their continuous support on the Lexus car. The authors acknowledge the contribution of the "Institut de Recherche Technologique NanoElec" which has been founded by the french program "Investissement d'Avenir" ANR-10-AIRT-05. Our thanks are also given to Nicolas Turro, Laurence Boissieux and Jean-François Cuniberto for their assistance in setting up the experiments.

## References

1. R. Labayrade, C. Royere, and D. Aubert. Experimental assessment of the rescue collision-mitigation system. *Vehicular Technology, IEEE Transactions on*, 56(1):89–102, Jan 2007.
2. N. Kaempchen, B. Schiele, and K. Dietmayer. Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 10(4), Jan 2009.
3. C. Laugier, I. Paromtchik, M. Perrollaz, Y. Mao, J-D Yoder, C. Tay, K. Mekhnacha, and A. Nègre. Probabilistic Analysis of Dynamic Scenes and Collision Risk Assessment to Improve Driving Safety. *Intelligent Transportation Systems Journal*, 3(4):4–19, November 2011.
4. A. Petrovskaya, M. Perrollaz, L. Oliveira, L. Spinello, R. Triebel, A. Makris, J-D. Yoder, U. Nunes, C. Laugier, and P. Bessière. Awareness of road scene participants for autonomous driving. In Azim Eskandarian, editor, *Handbook of Intelligent Vehicles*, pages 1383–1432. Springer, 2012.
5. Guillem Alenya, Amaury Nègre, and James L. Crowley. A Comparison of Three Methods for Measure of Time to Contact. In *IEEE Conference on Intelligent Robotics and Systems (IROS)*, St. Louis, États-Unis, October 2009.
6. MK Tay, Kamel Mekhnacha, Cheng Chen, and Manuel Yguel. An efficient formulation of the bayesian occupation filter for target tracking in dynamic environments. *International Journal of Vehicle Autonomous Systems*, 6(1):155–171, 2008.
7. A. Negre, L. Rummelhard, and C. Laugier. Hybrid sampling bayesian occupancy filter. In *IEEE International Symposium on Intelligent Vehicles*, 2014.
8. Tobias Gindele, Sebastian Brechtel, Joachim Schroder, and Rüdiger Dillmann. Bayesian occupancy grid filter for dynamic environments using prior map knowledge. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 669–676. IEEE, 2009.
9. Radu Danescu, Florin Oniga, and Sergiu Nedevschi. Modeling and tracking the driving environment with a particle-based occupancy grid. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1331–1342, 2011.
10. C. Laugier C. Fulgenzi, A. Spalanzani. Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grids. In *IEEE International Conference on Robotics and Automation*, 2007.
11. Juan David Adarve, Mathias Perrollaz, Alexandros Makris, and Christian Laugier. Computing Occupancy Grids from Multiple Sensors using Linear Opinion Pools. In *IEEE International Conference on Robotics and Automation*, St Paul, Minnesota, United States, May 2012.